US Army Corps of Engineers
Construction Engineering
Research Laboratories

USACERL Technical Manuscript 96/50
February 1996

Software Development Plan for the Assessment System for Aircraft Noise, Version 2.0

by
Kendra Z. Hoff and Eric T. Ohler

The Assessment System for Aircraft Noise (ASAN) is a computer system being developed to model the effects of subsonic and supersonic aircraft noise from Military Training Routes and Military Operations Areas. The purpose is to assist U.S. Air Force environmental and route planners in planning minimal impact routes and in producing improved environmental impact analysis documents.

This document provides a description of the activities to be performed by the U.S. Army Construction Engineering Research Laboratories during the design, development, and testing of the Computer Software Configuration Item for ASAN) Version 2.0. This information is necessary to guide project management of the ASAN Version 2.0 software development effort. The document also makes all management tasks and management information visible to the project sponsor and other interested parties.

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products. The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED

DO NOT RETURN IT TO THE ORIGINATOR

USER EVALUATION OF REPORT

REFERENCE: USACERL Technical Manuscript 96/50, *Software Development Plan for the Assessment System for Aircraft Noise, Version 2.0*

Please take a few minutes to answer the questions below, tear out this sheet, and return it to USACERL. As user of this report, your customer comments will provide USACERL with information essential for improving future reports.

1. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

____

____

____

2. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.)

____

____

3. Has the information in this report led to any quantitative savings as far as manhours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

____

____

4. What is your evaluation of this report in the following areas?

a. Presentation: ______________________

b. Completeness: ______________________

c. Easy to Understand: ______________________

d. Easy to Implement: ______________________

e. Adequate Reference Material: ______________________

f. Relates to Area of Interest: ______________________

g. Did the report meet your expectations? ______________________

h. Does the report raise unanswered questions? ______________________

i. General Comments. (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.)

____

____

____

____

____

____

5. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name: __________________________

Telephone Number: __________________________

Organization Address: __________________________

____

____

6. Please mail the completed form to:

Department of the Army
CONSTRUCTION ENGINEERING RESEARCH LABORATORIES
ATTN: CECER-TR-I
P.O. Box 9005
Champaign, IL 61826-9005

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE February 1996	3. REPORT TYPE AND DATES COVERED Final

4. TITLE AND SUBTITLE
Software Development Plan for the Assessment System for Aircraft Noise, Version 2.0

5. FUNDING NUMBERS
MIPR
FQ76249500067

6. AUTHOR(S)
Kendra Z. Hoff and Eric T. Ohler

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Construction Engineering Research Laboratories (USACERL)
P.O. Box 9005
Champaign, IL 61826-9005

8. PERFORMING ORGANIZATION REPORT NUMBER
TM 96/50

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Air Force Armstrong Laboratory
ATTN: AL/OEBN (NSBIT)
2610 Seventh Street, Bldg. 441
Wright-Patterson AFB, OH 45433-7901

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Copies are available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The Assessment System for Aircraft Noise (ASAN) is a computer system being developed to model the effects of subsonic and supersonic aircraft noise from Military Training Routes and Military Operations Areas. The purpose is to assist U.S. Air Force environmental and route planners in planning minimal impact routes and in producing improved environmental impact analysis documents.

This document provides a description of the activities to be performed by the U.S. Army Construction Engineering Research Laboratories during the design, development, and testing of the Computer Software Configuration Item for ASAN) Version 2.0. This information is necessary to guide project management of the ASAN Version 2.0 software development effort. The document also makes all management tasks and management information visible to the project sponsor and other interested parties.

14. SUBJECT TERMS
computer system
software
noise assessment procedures
environmental impact analysis

15. NUMBER OF PAGES
50

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT SAR

Foreword

This study was conducted for the U.S. Air Force Armstrong Laboratory, Noise and Sonic Boom Impact Technology Advanced Development Program Office (NSBIT) under Military Interdepartmental Purchase Request (MIPR) No. FQ76249500067; Work Unit PN6, “ASAN Software Development.” The technical monitor was MAJ Jeffrey L. Fordon, AL/OEBN (NSBIT).

The work was performed by the Engineering Processes Division (PL-E) of the Planning and Management Laboratory (PL), U.S. Army Construction Engineering Research Laboratories (USACERL). The USACERL principal investigator was Kendra Z. Hoff. Dr. Michael P. Case is Chief, CECER-PL-E; L. Michael Golish is Operations Chief, CECER-PL; and Dr. David M. Joncich is Chief, CECER-PL. The USACERL technical editor was Linda L. Wheatley, Technical Resources Center.

COL James T. Scott is Commander and Acting Director, and Dr. Michael J. O'Connor is Technical Director of USACERL.

Table of Contents

1 Scope 1
1.1 Identification 1
1.2 System Overview 1
1.3 Document Overview 1
1.4 Relationship to Other Plans 2

2 Referenced Documents 3
2.1 Specifications 3
2.2 Military Standards 3
2.3 Other Documents 3

3 Overview of Required Work 5
3.1 Requirements and Constraints on the Software To Be Developed . . 5
3.2 Requirements and Constraints on Project Documentation 5
3.3 Position of the Project in the System Life Cycle 5
3.4 The Selected Program/Acquisition Strategy 6
3.5 Requirements and Constraints on Project Schedules and Resources 6
3.6 Other Requirements and Constraints 6

4 Plans for Performing General Software Development Activities 7
4.1 Software Development Process 7
4.2 General Plans for Software Development 9
4.2.1 Software Development Methods 9
4.2.2 Standards for Software Products 9
4.2.3 Reusable Software Products 9
4.2.4 Handling of Critical Requirements 9
4.2.5 Computer Hardware Resource Utilization 9
4.2.6 Recording Rationale 9
4.2.7 Access for Acquirer Review 10

5 Plans for Performing Detailed Software Development Activities 11
5.1 Project Planning and Oversight 11
5.1.1 Software Development Planning 11
5.1.2 CSCI Test Planning 11
5.1.3 System Test Planning 11
5.1.4 Software Installation Planning 11
5.1.5 Software Transition Planning 11
5.1.6 Following and Updating Plans 11

5.2 Establishing a Software Development Environment 12

5.2.1 Software Engineering Environment 12

5.2.2 Software Test Environment 13

5.2.3 Software Development Library 13

5.2.4 Software Development Files 13

5.2.5 Non-Deliverable Software 14

5.3 System Requirements Analysis 14

5.4 System Design 14

5.5 Software Requirements Analysis 14

5.6 Software Design 14

5.6.1 CSCI-Wide Design Decisions 14

5.6.2 CSCI Architectural Design 15

5.6.3 CSCI Detailed Design 15

5.7 Software Implementation and Unit Testing 15

5.7.1 Software Implementation 15

5.7.2 Preparing for Unit Testing 15

5.7.3 Performing Unit Testing 16

5.7.4 Revision and Retesting 16

5.7.5 Analyzing and Recording Unit Test Results 16

5.8 Unit Integration and Testing 16

5.8.1 Preparing for Unit Integration and Testing 16

5.8.2 Performing Unit Integration and Testing 16

5.8.3 Revision and Retesting 17

5.8.4 Analyzing and Recording Unit Integration and Test Results 17

5.9 CSCI Qualification Testing 17

5.9.1 Independence in CSCI Qualification Testing 17

5.9.2 Testing on the Target Computer System 17

5.9.3 Preparing for CSCI Qualification Testing 17

5.9.4 Dry Run of CSCI Qualification Testing 18

5.9.5 Performing CSCI Qualification Testing 18

5.9.6 Revision and Retesting 18

5.9.7 Analyzing and Recording CSCI Qualification Test Results . 18

5.10 CSCI/HWCI Integration and Testing 18

5.11 System Qualification Testing 18

5.12 Preparing for Software Use 18

5.13 Preparing for Software Transition 19

5.14 Software Configuration Management 19

5.14.1 Configuration Identification 19

5.14.2 Configuration Control 20

5.14.3 Configuration Status Accounting 20

5.14.4 Configuration Audits 20
5.14.5 Packaging, Storage, Handling, and Delivery 21
5.15 Software Product Evaluation 21
5.16 Software Quality Assurance 21
5.16.1 Software Quality Assurance Evaluations 21
5.16.2 Software Quality Assurance Records 21
5.16.3 Independence in Software Quality Assurance 21
5.17 Corrective Action 22
5.17.1 Problem/Change Reports 22
5.17.2 Corrective Action System 22
5.18 Joint Technical and Management Reviews 22
5.18.1 Joint Technical Reviews 22
5.18.2 Joint Management Reviews 23
5.19 Other Software Development Activities 23
5.19.1 Risk Management 23
5.19.2 Software Management Indicators 23
5.19.3 Security and Privacy 24
5.19.4 Subcontractor Management 24
5.19.5 Interface With Software Independent Verification and Validation Agents 24
5.19.6 Coordination with Associate Developers 24
5.19.7 Improvement of Project Processes 24
5.19.8 Other Activities Not Covered Elsewhere in the Plan 24

6 Schedules and Activity Network 27

7 Project Organization and Resources 29
7.1 Project Organization 29
7.2 Project Resources 29

8 Notes 31
8.1 Acronyms 31

Appendix A: Problem/Change Report Form 33

Appendix B: Software Development Schedule 35

Appendix C: Data Development Schedule 39

Tables and Figures

Table I Structure of the Software Development Plan 2
Table II ASAN Version 2.0 Documents 5
Table III Document Submission Requirements 6
Table IV Build Elements 8
Table V Configuration Control Activity Numbers 19
Table VI Proposed Joint Technical Reviews 23
Table VII Project Personnel 28

Figure 1 ASAN Version 2.0 Software Development Process 7
Figure 2 Project Personnel Structure 29

1 Scope

1.1 Identification

This document provides a description of the activities to be performed by the U.S. Army Construction Engineering Research Laboratories (USACERL) during the design, development and testing of the Computer Software Configuration Item (CSCI) for the Assessment System for Aircraft Noise (ASAN) Version 2.0. This Software Development Plan (SDP) is identified in the Statement of Work (SOW) as CDRL Exhibit B, Sequence 10.

1.2 System Overview

ASAN is a computer system being developed to model the effects of subsonic and supersonic aircraft noise from Military Training Routes (MTRs) and Military Operations Areas (MOAs). The purpose is to assist U.S. Air Force environmental and route planners in planning minimal impact routes and in producing improved environmental impact analysis documents.

ASAN is sponsored by the U.S. Air Force Noise and Sonic Boom Impact Technology Advanced Development Program Office (NSBIT), which sponsors research supporting the preparation of scientifically supportable and legally defensible environmental assessments of the effects of aircraft noise on humans, animals, and structures. ASAN Version 1.0 was developed by Bolt Beranek and Newman Inc., Cambridge, MA, under Air Force Contracts F33615-90-D-0653 and F33615-86-C-0530. ASAN Version 2.0 will be developed by USACERL, Champaign, IL, under MIPR FQ76249500067.

1.3 Document Overview

This document serves two purposes. The first is to provide the information necessary to guide project management of the ASAN Version 2.0 software development effort. The second purpose is to make all management tasks and management information visible to the project sponsor and other interested parties.

All software for the ASAN Version 2.0 project will be developed according to this formal SDP. This plan is written in accordance with the U.S. Military Standard for Software Development and Documentation (MIL-STD-498) and the subsidiary Data

Item Definition (DID) DI-IPSC-81427, Software Development Plan. This SDP contains the sections shown in Table I as outlined by the chosen standard.

Table I Structure of the Software Development Plan

1. Scope
2. Referenced Documents
3. Overview of Required Work
4. Plans for Performing General Software Development Activities
5. Plans for Performing Detailed Software Development Activities
6. Schedules and Activity Network
7. Project Organization and Resources
8. Notes

1.4 Relationship to Other Plans

The ASAN Version 2.0 project is organized with the aid of two managerial documentation plans:

1. The Software Development Plan (SDP)—this document
2. The Software Test Plan (STP).

The SDP describes the approach to carrying out all software development activities, including an overview of the plan for testing the software. The STP expands on the testing plan defined in this SDP to describe the detailed approach to software qualification testing, including a description of the software test environment and identification of the tests to be performed.

2 Referenced Documents

2.1 Specifications

SOW, Assessment System for Aircraft Noise, 10 October 1995.

2.2 Military Standards

MIL-STD-498, *Software Development and Documentation*, 5 December 1994.

2.3 Other Documents

Reddingius, Nicolaas H., *Software Requirements Specification for the Beta Version of the Assessment System for Aircraft Noise*, BBN Report No. 7653 (Bolt Beranek and Newman Inc., 14 January 1992).

Reddingius, Nicolaas H., and Shawntise R. Turner, *Software User's Manual for the Assessment System for Aircraft Noise*, BBN Report No. 7730 (Bolt Beranek and Newman Inc., 6 February 1995).

Smyth, John S., Bruce Papazian, Nicolaas H. Reddingius, and Shawntise R. Turner, *Software Design Document for the Assessment System for Aircraft Noise*, BBN Report No. 7654 (Bolt Beranek and Newman Inc., 7 March 1995).

3 Overview of Required Work

3.1 Requirements and Constraints on the Software to be Developed

Each of the modifications identified in sections 3.1.1 - 3.1.4 of the SOW will be incorporated into ASAN Version 1.0 to create Version 2.0. The software will be developed according to the approach defined in this SDP.

3.2 Requirements and Constraints on Project Documentation

The documents shown in Table II will be produced to record the ASAN Version 2.0 software development effort.

Table II ASAN Version 2.0 Documents

DID	Title
DI-IPSC-81427/T	Software Development Plan (SDP) - this document
DI-IPSC-81438/T	Software Test Plan (STP)
DI-IPSC-81433/T	Software Requirements Specification (SRS)
DI-IPSC-81435/T	Software Design Description (SDD)
DI-IPSC-81439	Software Test Description (STD)
DI-IPSC-81441/T	Software Product Specification (SPS)
DI-IPSC-81442/T	Software Version Description (SVD)
DI-IPSC-81443/T	Software User Manual (SUM)

The existing SRS, SDD, and SUM for ASAN Version 1.0 will be updated to define the new ASAN Version 2.0 capabilities, conforming to the format and level of detail previously established in the documents. The SRS, SDD, and SUM will not be modified for any unchanged Version 1.0 capabilities. The remaining documents will be written by USACERL in accordance with MIL-STD-498 and the subsidiary DIDs.

3.3 Position of the Project in the System Life Cycle

ASAN is being developed using an evolutionary life-cycle model where the software product evolves in successively larger solutions. ASAN Version 1.0 is a functional software program currently being used by customers. ASAN Version 2.0 will be developed by incorporating modifications and enhancements into Version 1.0.

3.4 THE SELECTED PROGRAM/ACQUISITION STRATEGY

This paragraph has been tailored out.

3.5 REQUIREMENTS AND CONSTRAINTS ON PROJECT SCHEDULES AND RESOURCES

The work defined in the SOW will be completed for the amount of funding provided under MIPR FQ76249500067.

All deliverables must be provided to NSBIT no later than 12 months from the date of receipt of the SOW.

Each document will be provided to NSBIT adhering to the submission requirements shown in Table III.

Table III Document Submission Requirements

Document	Submission Requirement
SDP	within 45 days of receipt of the SOW
SRS	30 days prior to the architectural design review
SDD, architectural design	15 days prior to the architectural design review
STP	15 days prior to the first detailed design review
SDD, detailed design	15 days prior to a build's detailed design review
STD	30 days prior to a build's qualification test
SUM	30 days prior to a build's qualification test
SVD	30 days prior to a build's qualification test
SDD, as tested	within 15 days of a build's qualification test
SPS	within 30 days of the final qualification test

If modifications to a document are requested by NSBIT, the document will be resubmitted to NSBIT for approval within 15 days of the review or qualification test. The executable software and a final copy of all documents will be delivered to NSBIT within 30 days of the final qualification test.

3.6 OTHER REQUIREMENTS AND CONSTRAINTS

The ASAN Version 2.0 software will be developed in accordance with a tailored version of the MIL-STD-498 software development process as defined in this SDP.

The software shall operate under SunOS 4.1.3, Solaris 2.4, and IRIX 5.3 UNIX operating systems.

4 Plans for Performing General Software Development Activities

4.1 Software Development Process

The ASAN Version 2.0 software development process is shown in Figure 1. The documents that are prepared to record the results of each task are shown below each task box. The process consists of defining the software requirements, developing an architectural design, developing a detailed design, implementing and testing the software capabilities and delivering the software product.

During software requirements analysis, the software capabilities that are required for acceptance of ASAN Version 2.0 will be defined and recorded in the SRS. The architectural design will be documented in the SDD, which will include Data Flow Diagrams illustrating all control and data flow activity, interface specifications, and functional descriptions of the software components. The software requirements and the architectural design will be reviewed for approval at a Joint Management/Technical Review.

Following acceptance of the architectural design, the ASAN Version 2.0 modifications will be grouped into three build elements as shown in Table IV. The detailed design,

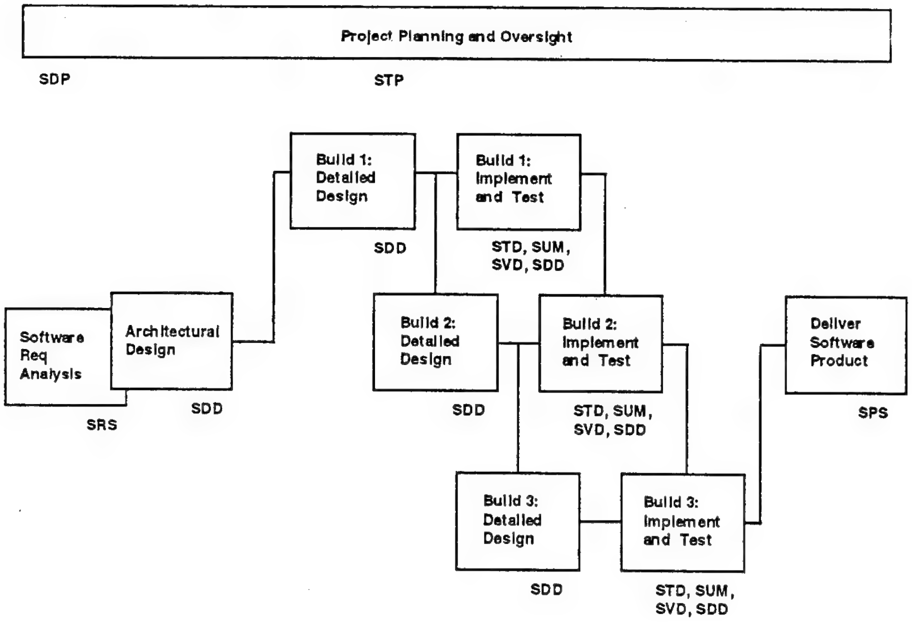


Figure 1 ASAN Version 2.0 Software Development Process

implementation, and testing of each build element will be scheduled and tracked separately. This approach will allow development to progress on an incremental basis with each build using the capabilities that were implemented and tested in previous builds.

Table IV Build Elements

Build	Modification	SOW Item
1	Select Assessment Area	3.1.1.a:1
1	Printer Control	3.1.1.d
1	Toggle/Radio Buttons	3.1.1.g
1	MTR Copy	3.1.1.b
1	Version 1.0 Bug Fixes	3.1.4
2	Import/Export Local Data	3.1.1.c
2	Auto ROI	3.1.1.f:1
2	Map Save	3.1.1.e
2	Version 1.0 Bug Fixes	3.1.4
3	PS.MAP & Postscript	3.1.3.a
3	Highlight Features in ROI	3.1.1.f:2
3	Import 1.0 Assessment	3.1.1.a:2
3	Version 1.0 Bug Fixes	3.1.4

The detailed design of each build element will describe how the design will be implemented, including complete descriptions of the software modules, database definitions, and interface details. The detailed design will be documented in the SDD. Each build element will be presented at detailed design level at a Joint Management/Technical Review. The approach that will be used for testing the software, recorded in the STP, will be reviewed in conjunction with the detailed design of Build 1.

When NSBIT approves a build element's detailed design, the build element will be implemented and tested by USACERL. The SUM will be updated to provide a guide for operating the software. A description of the test cases and test procedures to be used for the build element will be recorded in the STD. The SVD will identify which capabilities are included in the test version of the software.

Each build element will have a separately scheduled qualification test where the capabilities will be tested for approval by a NSBIT representative. Results of each qualification test will be recorded in the STD. The SDD will be updated to record the final design of the build element as it was tested and approved.

After the final build element has been tested and approved, the SPS will be prepared. It will contain or reference the executable software, source files, and software support information. Delivery of the software will be accompanied by a final version of all documents.

4.2 General Plans for Software Development

4.2.1 Software Development Methods

The methodology to be used for each software development activity is described in Section 5 of this SDP.

4.2.2 Standards for Software Products

The standards to be followed for representing requirements, design, code, test cases and test results are described in Sections 5.5 – 5.9 of this SDP.

4.2.3 Reusable Software Products

This paragraph has been tailored out.

4.2.4 Handling of Critical Requirements

This paragraph has been tailored out.

4.2.5 Computer Hardware Resource Utilization

ASAN Version 2.0 will be designed to operate under the minimum hardware requirements of Version 1.0. Design decisions that would increase the minimum requirements will be subject to the approval of NSBIT in a Joint Management/ Technical Review.

4.2.6 Recording Rationale

The rationale for decisions made on the project that have major impact, are controversial, or affect a large portion of the system will be recorded and maintained in the software development files.

4.2.7 Access for Acquirer Review

Representatives of NSBIT will have access to the USACERL software engineering and test environments for review of software products and activities on the dates of the Joint Management/Technical Reviews which are held at USACERL.

5 Plans for Performing Detailed Software Development Activities

5.1 Project Planning and Oversight

5.1.1 Software Development Planning

The results of software development planning are recorded in this SDP. If necessary or beneficial changes to this SDP are identified, an updated SDP will be submitted to NSBIT for approval.

5.1.2 CSCI Test Planning

The overall approach to CSCI test planning is described in Section 5.9 of this SDP. The detailed plan for qualification testing will be defined in the STP.

5.1.3 System Test Planning

This paragraph has been tailored out.

5.1.4 Software Installation Planning

This paragraph has been tailored out.

5.1.5 Software Transition Planning

This paragraph has been tailored out.

5.1.6 Following and Updating Plans

The software development activities for this project will be conducted in accordance with this SDP. The status of all activities will be evaluated in the Joint Management/

Technical Reviews to assure that each activity is being performed in accordance with the plan. Updates to this SDP will be subject to approval from NSBIT.

5.2 Establishing a Software Development Environment

5.2.1 Software Engineering Environment

A software development environment will be established in the Engineering Processes Division of USACERL. Since ASAN Version 2.0 will be developed by incorporating enhancements into existing source code and documentation, it will be most time and cost efficient to update the products with the tools used for Version 1.0. CASE tools will not be used. The environment will consist of:

- 4 Sun workstations running SunOS 4.1.3
- 1 Sun workstation running Solaris 2.4
- 1 SGI workstation running IRIX 5.3
- X Window System
- Motif Developer Pack
- GRASS 4.1 source code and executable programs
- Oracle 7 developer package and Pro*C preprocessor
- ASAN 1.0 source code and executable programs
- SCCS configuration control software
- Compilers and related utilities: gcc (Sun), cc (SGI), make, ld, ranlib, awk, sed, perl, C code libraries and include files
- Text and code editors: vi, emacs, vim
- Debuggers: dbx, ddd, gdb
- 3 DOS-based computers
- WordPerfect for Windows 5.2
- PVCS configuration control software.

A data development environment will be established in the Resource Mitigation and Protection Division of USACERL consisting of:

- Sun workstations
- DOS-based computers
- Arc/Info release 7.1 (TIN,COGO,GRID extensions)
- Excel
- Oracle release 7.1.x
- ArcView

- Pageview, Ghostview
- AutoCad
- Wingz spreadsheet
- GRASS release 4.1
- Internet tools (Mosaic, lynx, ftp)
- GRASS contrib libraries
- WordPerfect 5.1
- Proj libraries.

5.2.2 Software Test Environment

The software test environment will be a subset of the software development environment described in Section 5.2.1 of this SDP. It consists of:

- 1 Sun workstation running SunOS 4.1.3
- 1 Sun workstation running Solaris 2.4
- 1 SGI workstation running IRIX 5.3
- X Window System
- Motif Window Manager
- GRASS 4.1 executable programs
- Oracle 7 executable programs
- ASAN executable programs.

5.2.3 Software Development Library

The software that is produced for this project will be maintained in a software development library under the SunOS 4.1.3 operating system. The software library will be controlled using SCCS configuration control software. The project documents will be maintained in a documentation library on a DOS-based server using PVCS Windows configuration control software. The control and management of these libraries is described in Section 5.14 of this SDP.

5.2.4 Software Development Files

The software development team leader will maintain software development files. These files will contain the SOW, ASAN correspondence, Joint Management/Technical Review notes and summary reports, status reports, problem/change reports, rationale for key decisions, test cases and results, and a copy of all documents prepared for this project.

5.2.5 Non-Deliverable Software

Non-deliverable software will be developed and maintained by the data development team for specific data conversion tasks.

5.3 SYSTEM REQUIREMENTS ANALYSIS

This paragraph has been tailored out.

5.4 SYSTEM DESIGN

This paragraph has been tailored out.

5.5 SOFTWARE REQUIREMENTS ANALYSIS

The SRS will be created by modifying the *Software Requirements Specification for the Beta Version of the Assessment System for Aircraft Noise* to include descriptions of the software characteristics that are required for acceptance of ASAN Version 2.0. Changes to the SRS will conform to the format and level of detail previously established in the document. The SRS will be reviewed for approval by NSBIT at a Joint Management/Technical Review.

5.6 SOFTWARE DESIGN

5.6.1 CSCI-Wide Design Decisions

Any CSCI-wide design decisions will be recorded in the SDD and presented for review at a Joint Management/Technical Review.

5.6.2 CSCI Architectural Design

The architectural design will be performed using the Ward/Mellor design methodology. The existing ASAN SDD will be modified to include the architectural design of ASAN 2.0 capabilities. It will include Data Flow Diagrams illustrating all control and data flow activity, interface specifications and functional descriptions of the software components. Changes to the SDD will conform to the format and level of detail previously established in the document. The architectural design portion of the SDD will be reviewed for approval by NSBIT at a Joint Management/Technical Review.

5.6.3 CSCI Detailed Design

The detailed design of ASAN 2.0 capabilities will be performed using the Ward/Mellor design methodology. It will be recorded in the SDD, including complete descriptions of the software modules, database definitions, and interface details. Changes to the SDD will conform to the format and level of detail previously established in the document. The detailed design portion of the SDD will be reviewed for approval by NSBIT at a Joint Management/Technical Review.

5.7 SOFTWARE IMPLEMENTATION AND UNIT TESTING

5.7.1 Software Implementation

The approved detailed designs will be implemented in the ASAN source code. The implementation of ASAN Version 2.0 capabilities will conform to the coding standards previously established in Version 1.0 to ensure consistency of the code. The code will be implemented under the SunOS 4.1.3 operating system and will then be ported to the Solaris 2.4 and IRIX 5.3 operating systems.

5.7.2 Preparing for Unit Testing

The preparation of unit test cases will be the responsibility of the programmer who implemented an ASAN Version 2.0 modification. For each software unit, test cases will be developed using structural testing methods, based on the structure of the code, and functional testing methods, based on the functional specifications. The test cases will be recorded and maintained in the software development files.

5.7.3 Performing Unit Testing

For each ASAN Version 2.0 modification, the affected software units will be tested by the programmer in accordance with the unit test cases.

5.7.4 Revision and Retesting

Based on the results of unit testing, any necessary revisions to the software will be performed by the programmer. The software unit will be retested, and the software documentation will be updated as needed. This procedure will be repeated until all test cases have been satisfied.

5.7.5 Analyzing and Recording Unit Test Results

The software development team leader will evaluate and approve the results of unit testing. The test results will be recorded with the test cases and maintained in the software development files.

5.8 Unit Integration and Testing

5.8.1 Preparing for Unit Integration and Testing

A software team member who was not involved in the detailed design and implementation will be responsible for preparing integration test cases and performing integration testing. For each build element, test cases will be developed using functional testing methods. The test cases will be recorded and maintained in the software development files.

5.8.2 Performing Unit Integration and Testing

Upon approval by the team leader of unit testing of an ASAN Version 2.0 modification, the affected software units will be integrated into the test version of the build element. The build element will be tested by the integration tester in accordance with the test cases.

5.8.3 Revision and Retesting

Any problems detected during integration testing will be recorded on a problem/change report and submitted to the team leader. Necessary revisions to the software will be performed by the programmer. The software will be retested and the software documentation will be updated as needed. This procedure will be repeated until all test cases have been satisfied.

5.8.4 Analyzing and Recording Unit Integration and Test Results

The team leader will evaluate and approve the results of integration testing. The test results will be recorded with the test cases and maintained in the software development files.

5.9 CSCI QUALIFICATION TESTING

5.9.1 Independence in CSCI Qualification Testing

A NSBIT representative will be responsible for performing qualification testing of each build element.

5.9.2 Testing on the Target Computer System

Each build element will be tested on the target computer systems, including the SunOS 4.1.3, Solaris 2.4 and IRIX 5.3 Unix operating systems. The software test environment will be further defined in the STP.

5.9.3 Preparing for CSCI Qualification Testing

Preparation for qualification testing of each build element will be performed by a member of the software team who was not involved in the detailed design and implementation. Test cases that determine whether all requirements have been met will be developed for each build element using functional testing methods. Necessary test data will be prepared. The test cases will be recorded in the STD. The date of qualification testing for each build element will be scheduled with NSBIT.

5.9.4 Dry Run of CSCI Qualification Testing

A dry run of the test cases will be performed before the qualification test for a build element to ensure that the software is ready for qualification testing.

5.9.5 Performing CSCI Qualification Testing

For each build element, a NSBIT representative will test the software in accordance with the qualification test cases.

5.9.6 Revision and Retesting

Any problems detected during qualification testing will be recorded on a problem/change report and submitted to the software development team leader. Necessary revisions to the software will be performed by the programmer. The software will be retested by the development team and the software documentation will be updated as needed. Another qualification test will be scheduled with NSBIT. This procedure will be repeated until all test cases have been satisfied.

5.9.7 Analyzing and Recording CSCI Qualification Test Results

NSBIT will evaluate and approve the results of qualification testing. The test results will be recorded with the test cases in the STD.

5.10 CSCI/HWCI Integration and Testing

This paragraph has been tailored out.

5.11 System Qualification Testing

This paragraph has been tailored out.

5.12 Preparing for Software Use

This paragraph has been tailored out.

5.13 Preparing for Software Transition

This paragraph has been tailored out.

5.14 Software Configuration Management

5.14.1 Configuration Identification

Each controlled file (program files and documentation) in the software development libraries will be assigned a unique identifier. The identification scheme is:

filename-(activity #)(submission #).(revision #)

where the activity number indicates the software development activity for which the item is being submitted for approval, corresponding to the numbers shown in Table V.

Table V Configuration Control Activity Numbers

Activity	Number
Planning	1
Requirements Analysis	2
Architectural Design	3
Detailed Design, Build 1	4
Detailed Design, Build 2	5
Detailed Design, Build 3	6
Implementation, Build 1	7
Implementation, Build 2	8
Implementation, Build 3	9
Product Delivery	10

For example, filex-72.4 indicates the fourth revision of the file named filex for the second submission in the implementation of Build 1. The revision numbers will only be used within the USACERL software development team. To someone outside USACERL, the identification number of the first submission of this SDP is SDP-11.

The identification numbers will be maintained by the configuration control software. Document identifiers will be shown on each page of the printed document.

5.14.2 Configuration Control

The software development libraries will be controlled through the use of configuration control software that supports check in and check out capabilities, change tracking, access control, and archiving. The software development team leader will be responsible for authorizing and determining priorities and schedules of all changes. Enhancements to be incorporated into the software and the preparation of required documents will be authorized based on the schedule shown in Appendix B of this SDP. Changes submitted through the corrective action procedure, described in Section 5.17 of this SDP, will be scheduled based on priority and category.

The software development team leader will assign access rights to the files for each team member. All change activities will be documented through the configuration control software. Files will be marked to identify revisions that have been submitted and accepted.

5.14.3 Configuration Status Accounting

Records of the configuration status of each item under configuration control will be maintained in the following reports in the software development files:

- The monthly status reports, described in Section 5.19.2 of this SDP, contain the status of all software development activities and the status of all problem/change reports.

- The SVD, submitted before each qualification test, lists all software and documentation comprising the current version.

Additional configuration status reports are available when needed through the configuration control software.

5.14.4 Configuration Audits

Formal audits are not required for this project. The software development activities will be evaluated in the Joint Management/Technical Reviews.

5.14.5 Packaging, Storage, Handling, and Delivery

All controlled files are stored in the software development libraries. Daily backups will be made to tape. Baselines of the initial version and the latest accepted version of each file will be archived on a separate tape. The software development team leader will authorize copies made for submission to NSBIT.

5.15 SOFTWARE PRODUCT EVALUATION

This paragraph has been tailored out.

5.16 SOFTWARE QUALITY ASSURANCE

5.16.1 Software Quality Assurance Evaluations

The status of all software development activities will be evaluated during the Joint Management/Technical Reviews to assure that each required activity is being performed in accordance with this SDP and that each software product exists and has undergone evaluations and testing as defined by this SDP.

5.16.2 Software Quality Assurance Records

Records of each software quality assurance activity will be maintained in the software development files. These records will include status reports, Joint Management/ Technical Review notes and summary reports, problem/change reports, test cases and results, and a copy of all software documents prepared for this project.

5.16.3 Independence in Software Quality Assurance

NSBIT will conduct software quality assurance evaluations during each Joint Management/Technical Review. Problems in software products or activities required by the SOW or described in this SDP will be handled as described in Section 5.17.

5.17 Corrective Action

5.17.1 Problem/Change Reports

The problem/change report form in Appendix A will be used to describe each problem detected in the software and each problem in activities described in this SDP or required by the SOW. These reports will serve as input to the corrective action system.

5.17.2 Corrective Action System

The corrective action system will be initiated when a problem/change report is submitted to the software development team leader. Within 7 days of receiving a problem/change report, the team leader will initiate action for resolving the problem. Each problem will be classified by category and priority, using the categories and priorities in Appendix C of MIL-STD-498. Actions taken to resolve a problem will be recorded on the problem/change report. The status of all problems will be described in the monthly status reports. The problem/change reports will be maintained in the software development files.

5.18 Joint Technical and Management Reviews

5.18.1 Joint Technical Reviews

Joint Technical Reviews will be used to demonstrate and review proposed technical solutions, surface and resolve technical issues, and evaluate the evolving software products, using the software product evaluation criteria in Appendix D of MIL-STD-498. Each review will be attended by persons with technical knowledge of the software products to be reviewed. The reviews will focus on in-process and final software products, rather than materials generated especially for the review. Table VI lists a proposed set of reviews, including the software products to be evaluated in each review.

Table VI Proposed Joint Technical Reviews

Date	Product(s) Evaluated
28 Nov 95	SRS and Architectural Design SDD
01 Feb 96	STP and Build 1 Detailed Design SDD
21 Feb 96	Build 2 Detailed Design SDD
16 Apr 96	Build 3 Detailed Design SDD and Build 1 Qualification Test, STD, SUM, and SVD
31 May 96	Build 2 Qualification Test, STD, SUM and SVD
08 Jul 96	Build 3 Qualification Test, STD, SUM and SVD

For each software product being evaluated for acceptance, NSBIT will decide whether to accept the product without further modification, reject the product due to severe errors (once corrected, another review will be conducted), or accept the product provisionally (minor errors must be corrected). Problem/change reports will be used to record any problems that are detected in the software products.

5.18.2 Joint Management Reviews

Joint Management Reviews will be held in conjunction with the Joint Technical Reviews. The purposes of the management reviews are to inform NSBIT of the overall status of the software development activities and to resolve management-level issues and risks. The management portion of each review will be attended by persons with the authority to make cost and schedule decisions. A status report, as described in Section 5.19.2 of this SDP, will be provided to NSBIT 7 days before each review.

5.19 Other Software Development Activities

5.19.1 Risk Management

Identified risks will be included in the monthly status reports, described in Section 5.19.2 of this SDP. Mitigation strategies will be discussed in Joint Management/ Technical Reviews.

5.19.2 Software Management Indicators

Software management indicators will be used to aid in managing the software development process and in communicating its status to NSBIT. The indicators will be recorded in a status report that will be submitted to NSBIT 7 days before each Joint Management/Technical Review. The indicators to be used include:

- Accomplishments in the current reporting period
- Schedule status: percent complete of each of the tasks listed in the schedule in Appendix B of this SDP
- Problem/change report status: total number, number closed, number opened in the current reporting period, and priority and category of open problems
- Milestone performance: planned and actual dates of project milestones
- Identification of current and anticipated issues and risks.

5.19.3 Security and Privacy

This paragraph has been tailored out.

5.19.4 Subcontractor Management

This paragraph has been tailored out.

5.19.5 Interface with Software Independent Verification and Validation Agents

This paragraph has been tailored out.

5.19.6 Coordination with Associate Developers

This paragraph has been tailored out.

5.19.7 Improvement of Project Processes

When any necessary or beneficial improvements to the software development processes are identified, proposed updates to this SDP will be submitted to NSBIT for approval.

5.19.8 Other Activities Not Covered Elsewhere in the Plan

The ASAN Version 1.0 data will be updated for Version 2.0. The cartographic data updates for ASAN GRASS data involve the following layers: state and country boundaries, commercial air traffic corridors, MOAs, national parks, wilderness areas, wildlife refuges, interstate and federal highways, rivers, cities and towns, civilian airports, military air bases, and elevations. Data will also be converted and updated bimonthly for MTRs. The data development effort involves the following tasks:

- Three technical tasks will be accomplished to allow the user to determine the geographic region where ASAN calculations will be performed without having to conform to a tile boundary. These tasks are to determine: (1) a projection, (2) a coordinate system, and (3) a data storage mechanism (tiling scheme) to be used for all data layers. The projection and coordinate system will require data to be used without being limited to a specific location; however, CONUS and Alaska will be considered as two separate areas. This work will be completed before beginning the software design so the results can be used in the design.
- The data for each layer will be revised by locating current sources for each type of data. Each layer will then be converted to the proper projection and formatted to fit into the storage mechanism. Each layer will be converted individually, as the original data format will vary with the source of the data. This work will be completed before beginning the implementation of the first build so that the data can be used for testing.
- MTR updates will be produced and delivered to NSBIT bimonthly. As new MTR data is received, the current data will be examined for changes, additions, and deletions. The new data must also conform to the projection system and tiling scheme being used.

6 Schedules and Activity Network

The schedule of ASAN Version 2.0 software development activities is shown in Appendix B. The data development schedule is in Appendix C.

7 PROJECT ORGANIZATION AND RESOURCES

7.1 PROJECT ORGANIZATION

NSBIT is the ASAN Version 2.0 sponsor and acquirer. The project will be developed by USACERL under MIPR FQ76249500067. Within USACERL, the software development will be performed by the Planning and Management Laboratory, Engineering Processes Division. The data will be developed by the Land Management Laboratory, Resource Mitigation and Protection Division. Project management will be conducted through the Land Management Laboratory.

7.2 PROJECT RESOURCES

All development activities will be performed at USACERL in Champaign, IL using the hardware and software items acquired for this project and described in Section 5.2 of this SDP. The work will be accomplished by eight software professionals as depicted in Figure 2.

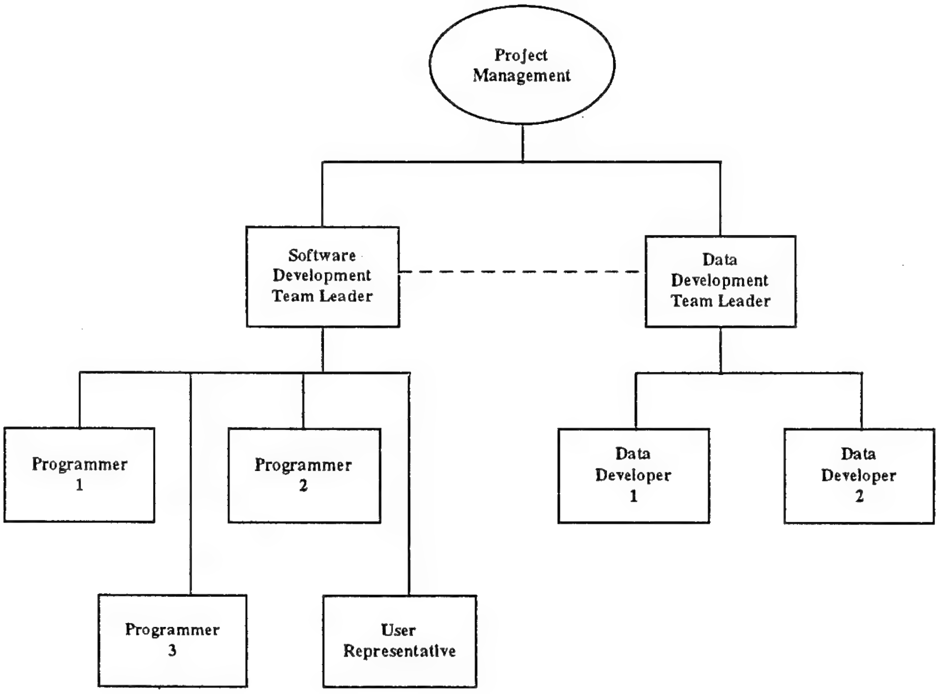


Figure 2 Project Personnel Structure

The work has been divided into two efforts: software development (SD) and data development (DD). Coordination of the two efforts will be the responsibility of the SD team leader and the DD team leader. The teams are composed of the personnel shown in Table VII.

Table VII Project Personnel

Role	Name	% Time	Responsibilities
SD Team Leader	Kendra Hoff	100	Coordinate, direct, and review the work of the SD team, perform administrative tasks, and report the team status to the sponsor
Programmer 1	David Stigberg	75	Design, implement and unit test the software
Programmer 2	Chenping Ni	50	Design, implement and unit test the software
Programmer 3	Tao Ning	50	Design, implement and unit test the software
User Representative	Sara Ort	100	Perform integration testing and prepare the software test plan, qualification test cases and the user manual
DD Team Leader	Eric Ohler	50	Coordinate, direct, and review the work of the DD team, perform administrative tasks, report the team status to the sponsor, and perform data development
Data Developer 1	Marcus Tooze	100	Perform data development
Data Developer 2	Mark Gibb	25	Perform data development

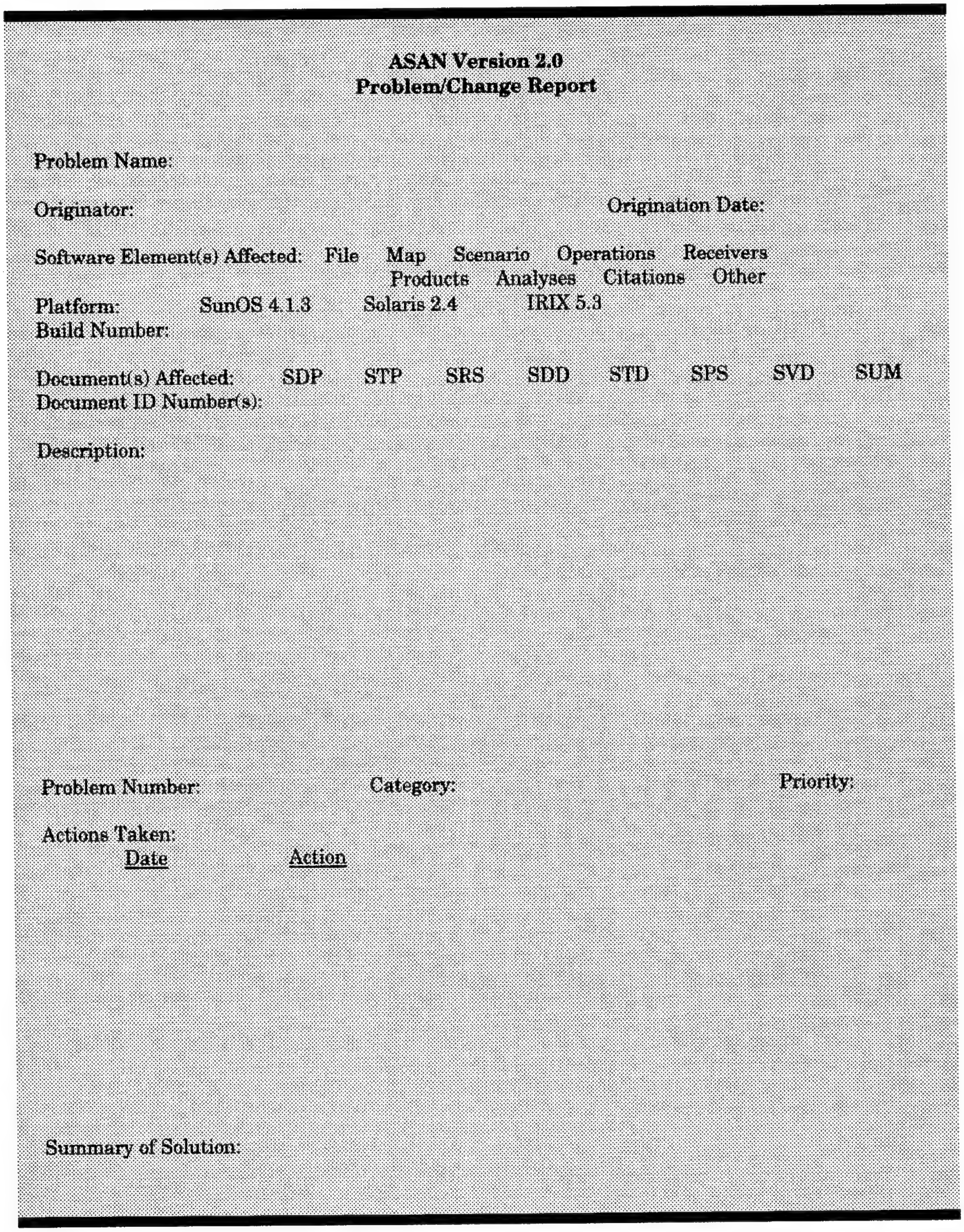
NSBIT has provided the Version 1.0 software, documentation, geographic database, and the original MTR release to USACERL. No additional resources are required from NSBIT to accomplish this work.

8 Notes

8.1 Acronyms

ASAN	Assessment System for Aircraft Noise
CSCI	Computer Software Configuration Item
DID	Data Item Definition
MOA	Military Operations Area
MTR	Military Training Route
NSBIT	Noise and Sonic Boom Impact Technology Advanced Development Program Office
SDD	Software Design Description
SDP	Software Development Plan
SOW	Statement of Work
SPS	Software Product Specification
SRS	Software Requirements Specification
STD	Software Test Description
STP	Software Test Plan
SUM	Software User Manual
SVD	Software Version Description
USACERL	U.S. Army Construction Engineering Research Laboratories

APPENDIX A: PROBLEM/CHANGE REPORT FORM



ASAN Version 2.0
Problem/Change Report

Problem Name:

Originator: Origination Date:

Software Element(s) Affected: File Map Scenario Operations Receivers
Products Analyses Citations Other

Platform: SunOS 4.1.3 Solaris 2.4 IRIX 5.3
Build Number:

Document(s) Affected: SDP STP SRS SDD STD SPS SVD SUM
Document ID Number(s):

Description:

Problem Number: Category: Priority:

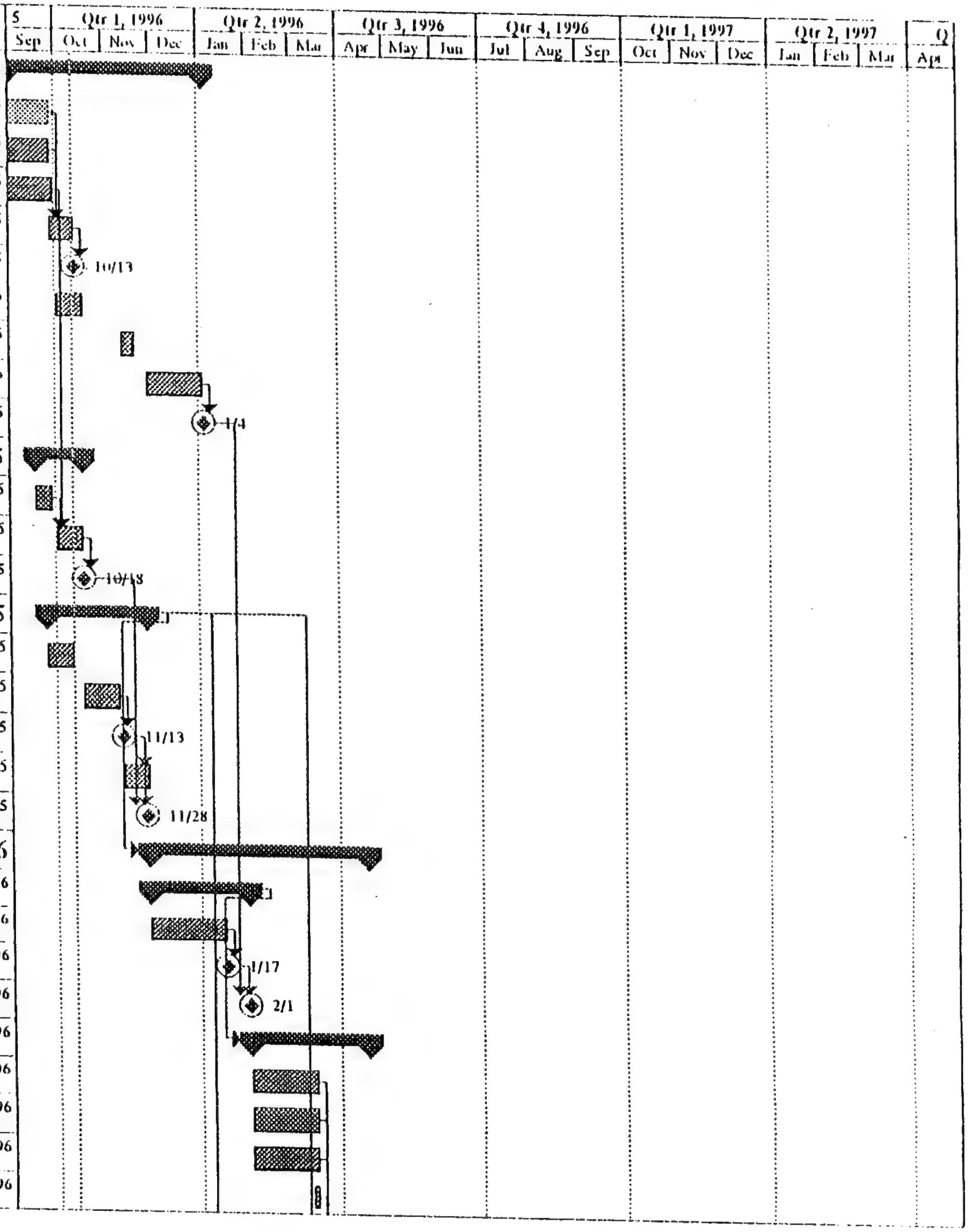
Actions Taken:
Date Action

Summary of Solution:

Appendix B: Software Development Schedule

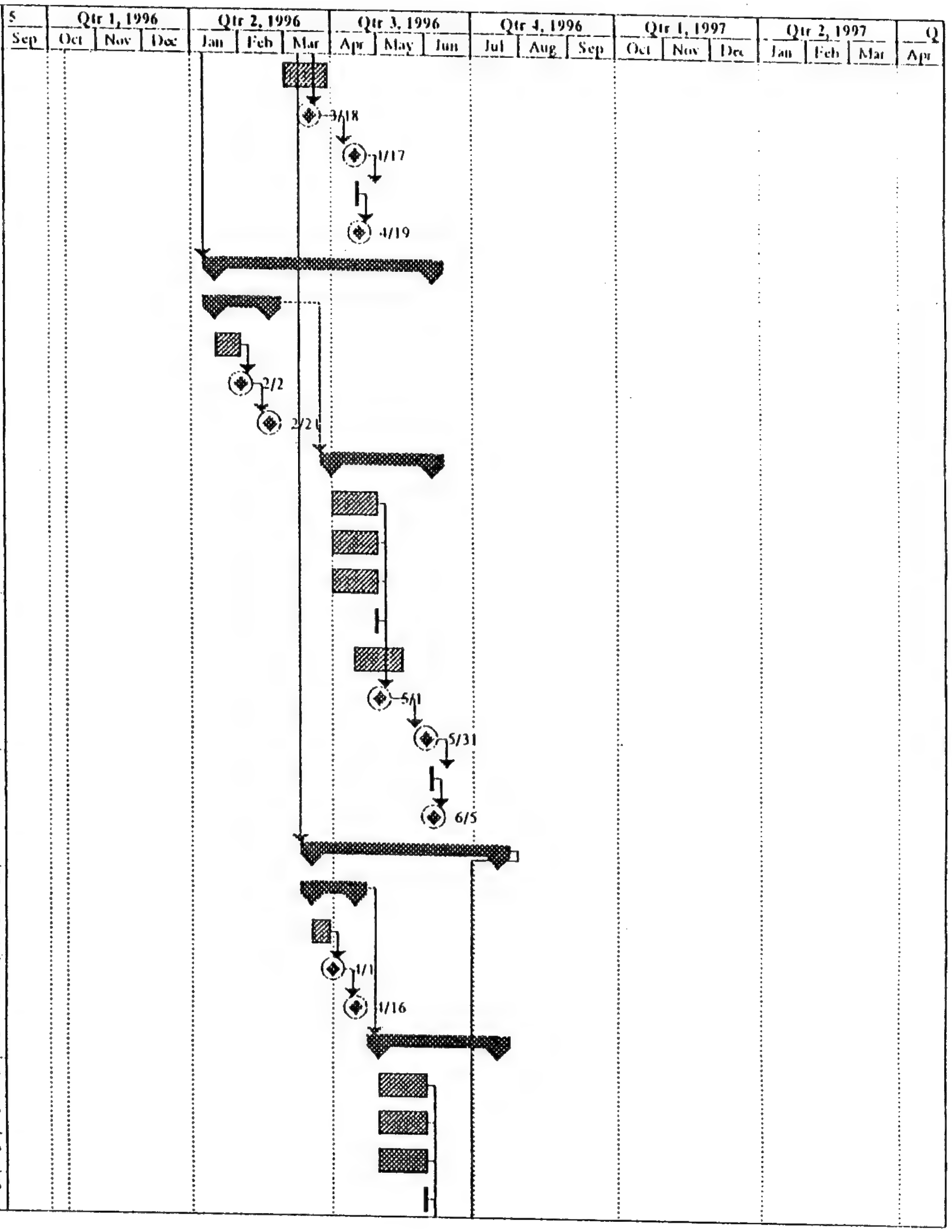
ASAN 2.0 Software Development
10/12/95

ID	Task Name	Start	Finish
1	**Planning**	**9/1/95**	**1/4/96**
2	Revise Statement of Work	9/1/95	9/27/95
3	Plan & schedule work	9/1/95	9/27/95
4	Set up SunOS 4.1.3 environment	9/1/95	9/29/95
5	Prepare SDP	9/28/95	10/12/95
6	Submit SDP	10/13/95	10/13/95
7	Prepare estimates of bug fixes	10/2/95	10/18/95
8	Set up Solaris 2.4 and IRIX 5.3	11/13/95	11/20/95
9	Prepare Software Test Plan	11/29/95	1/3/96
10	Submit STP	1/4/96	1/4/96
11	**Requirements Analysis**	**9/18/95**	**10/18/95**
12	Create electronic SRS	9/18/95	9/28/95
13	Prepare SRS	10/2/95	10/17/95
14	Submit SRS	10/18/95	10/18/95
15	**Architectural Design**	**9/25/95**	**11/28/95**
16	Define data structure for Eric	9/25/95	10/11/95
17	Prepare SDD at AD level	10/18/95	11/9/95
18	Submit SDD	11/13/95	11/13/95
19	Prepare for AD review	11/13/95	11/28/95
20	Architectural Design Review	11/28/95	11/28/95
21	**Build #1**	**11/29/95**	**4/19/96**
22	Detailed Design of Build #1	11/29/95	2/1/96
23	Prepare SDD at DD level	11/29/95	1/16/96
24	Submit SDD	1/17/96	1/17/96
25	Detailed Design Review	2/1/96	2/1/96
26	Implement Build #1	2/2/96	4/19/96
27	Code, test and update SDD	2/2/96	3/15/96
28	Prepare STD	2/2/96	3/15/96
29	Prepare SUM & Help files	2/2/96	3/15/96
30	Prepare SVD	3/13/96	3/15/96

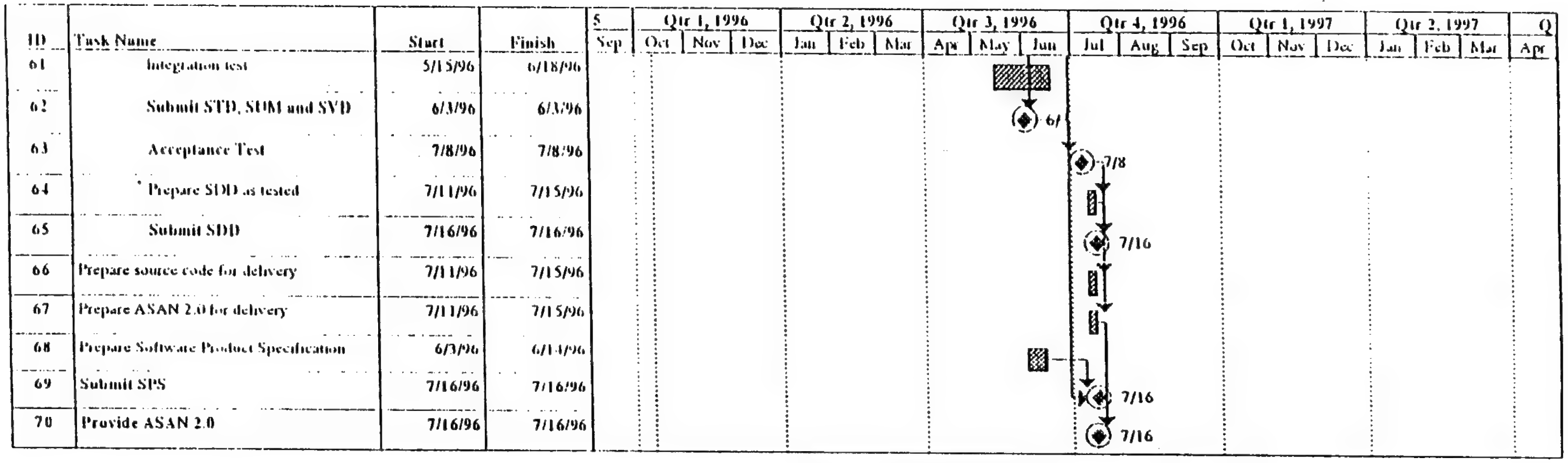


ASAN 2.0 Software Development
10/12/95

ID	Task Name	Start	Finish
31	Integration test	3/1/96	3/29/96
32	Submit STD, SUM and SVD	3/18/96	3/18/96
33	Acceptance Test	4/17/96	4/17/96
34	Prepare SDD as tested	4/17/96	4/18/96
35	Submit SDD	4/19/96	4/19/96
36	**Build #2**	**1/17/96**	**6/5/96**
37	Detailed Design of Build #2	1/17/96	2/21/96
38	Prepare SDD at DD level	1/17/96	2/1/96
39	Submit SDD	2/2/96	2/2/96
40	Detailed Design Review	2/21/96	2/21/96
41	Implement Build #2	4/1/96	6/5/96
42	Code, test and update SDD	4/1/96	4/30/96
43	Prepare STD	4/1/96	4/30/96
44	Prepare SUM & Help files	4/1/96	4/30/96
45	Prepare SVD	4/29/96	4/30/96
46	Integration test	4/15/96	5/16/96
47	Submit STD, SUM and SVD	5/1/96	5/1/96
48	Acceptance Test	5/31/96	5/31/96
49	Prepare SDD as tested	6/3/96	6/4/96
50	Submit SDD	6/5/96	6/5/96
51	**Build #3**	**3/18/96**	**7/16/96**
52	Detailed Design of Build #3	3/18/96	4/16/96
53	Prepare SDD at DD level	3/18/96	3/29/96
54	Submit SDD	4/1/96	4/1/96
55	Detailed Design Review	4/16/96	4/16/96
56	Implement Build #3	5/1/96	7/16/96
57	Code, test and update SDD	5/1/96	5/31/96
58	Prepare STD	5/1/96	5/31/96
59	Prepare SUM & Help files	5/1/96	5/31/96
60	Prepare SVD	5/30/96	5/31/96



ASAN 2.0 Software Development
10/12/95

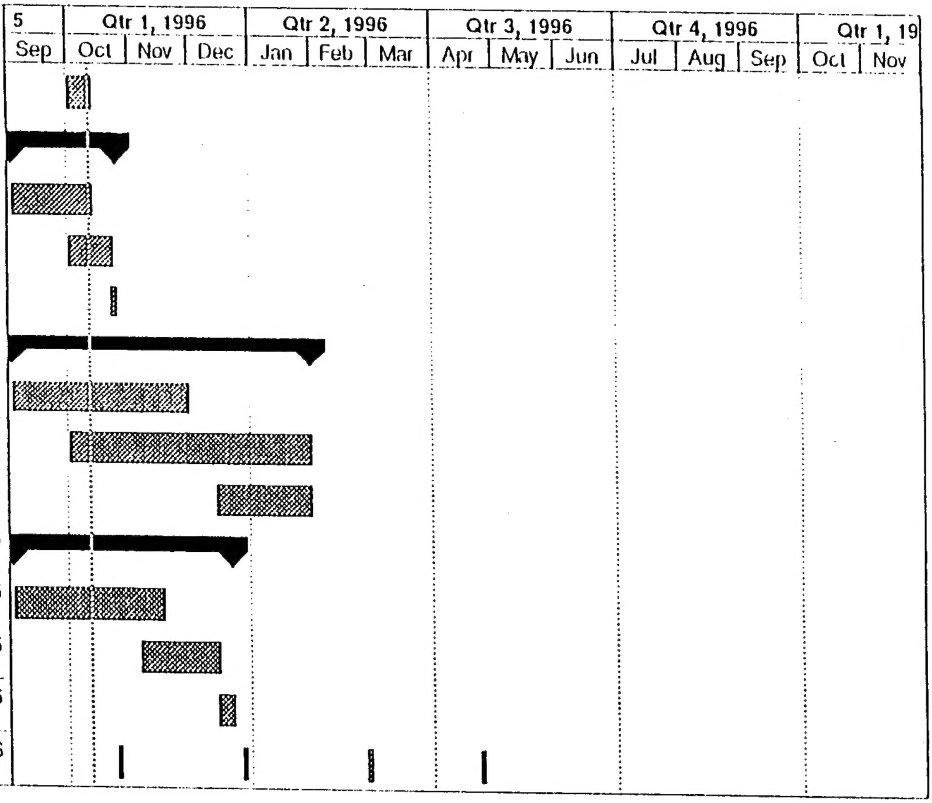


ID	Task Name	Start	Finish
61	Integration test	5/15/96	6/18/96
62	Submit STD, SUM and SVD	6/3/96	6/3/96
63	Acceptance Test	7/8/96	7/8/96
64	Prepare SDD as tested	7/11/96	7/15/96
65	Submit SDD	7/16/96	7/16/96
66	Prepare source code for delivery	7/11/96	7/15/96
67	Prepare ASAN 2.0 for delivery	7/11/96	7/15/96
68	Prepare Software Product Specification	6/3/96	6/14/96
69	Submit SPS	7/16/96	7/16/96
70	Provide ASAN 2.0	7/16/96	7/16/96

APPENDIX C: DATA DEVELOPMENT SCHEDULE

ASAN 2.0 Data Development
10/12/95

ID	Task Name	Start	Finish
1	Determine storage system, projection,	10/2/95	10/13/95
2	**Elevation (CONUS, Alaska)**	9/4/95	10/25/95
3	Determine source, acquisition	9/4/95	10/13/95
4	Data conversion and formatting	10/2/95	10/23/95
5	Create metadata	10/23/95	10/25/95
6	**Vector Layers (CONUS, Alaska)**	9/4/95	1/31/96
7	Determine sources, acquistion	9/4/95	11/30/95
8	Data conversion and formatting	10/2/95	1/31/96
9	Create metadata	12/15/95	1/31/96
10	**Site Lists (CONUS, Alaska)**	9/4/95	12/22/95
11	Determine sources, acquisition	9/4/95	11/17/95
12	Conversion, formatting	11/6/95	12/15/95
13	Create Metadata	12/15/95	12/22/95
14	MTR updates	10/25/95	4/25/96



DISTRIBUTION

Chief of Engineers 20314-1000
ATTN: CEHEC-IM-LH (2)
ATTN: CEHEC-IM-LP (2)
ATTN: CERD-L

Armstrong Laboratory 45433-7901
ATTN: AL/OEBN (NSBIT) (2)

Defense Tech. Info Center 22060-6218
ATTN: DTIC-O (2)

9
2/96

DEPARTMENT OF THE ARMY
CONSTRUCTION ENGINEERING RESEARCH LABORATORIES
CORPS OF ENGINEERS
PO BOX 9005
CHAMPAIGN, ILLINOIS 61826-9005

OFFICIAL BUSINESS